MARKOS GOGOULOS AND
DIOMIDIS SPINELLIS

using Linux live CDs for penetration testing

Markos Gogoulos is a research assistant in the ELTRUN Software Engineering and Security Group at the Athens University of Economics and Business and a free software movement enthusiast.

mgogoulos@gmail.com

Diomidis Spinellis is an associate professor in the Department of Management Science and Technology at the Athens University of Economics and Business and author of the books *Code Reading: The Open Source Perspective* (Addison-Wesley, 2003) and *Code Quality: The Open Source Perspective* (Addison-Wesley, 2006).

dds@aueb.gr

WHAT WOULD YOU THINK IF IN minutes you could have a full Linux system with almost all the necessary tools for penetration testing and security auditing, without having to install it on a dedicated machine? Whether you are a security professional or a system administrator, a bootable Linux live CD can be your best friend.

What Is Penetration Testing?

Penetration testing is a focused attempt to look for security holes. These can be design weaknesses or technical flaws and vulnerabilities in critical resources for a network. The test focuses on a network's infrastructure, servers, and workstations. Penetration testers try to break into a network, attempting to locate and document all security flaws, so that they can be fixed. Usually penetration testers are supplied with specific instructions as to which systems and networks to test. If you are to undertake such an effort, make sure you obtain written permission from a person authorized to give it, before even preparing for the test. Also notify all system administrators whose systems will be affected, because the test may create a heavy traffic load on the network and generate intrusion-detection system alerts. Penetration testing is quite similar to hacking—that's why it is also called ethical hacking—but differs in that it is arranged and approved by the network's owner and aims at locating all security flaws. This contrasts with hacking, where the goal is typically to find a single series of flaws that is sufficient for system intrusion. Whereas in hacking creativity has a major impact on the results and an instinctive, probably self-developed procedure is being followed, professional penetration testing involves the use of a methodology that will be followed to assure that results are accurate and complete.

The Need for a Methodology

A penetration testing methodology provides a framework that is followed to ensure that the results will be accurate and complete. As far as we know, the only publicly available methodology for penetration testing is the *Open Source Security Testing Methodology Manual* (OSSTMM) [1]. As quoted on OSSTMM's site:

> The OSSTMM is a peer-reviewed methodology for performing security tests and metrics. The OSSTMM test cases are divided

into five channels (sections) which collectively test: information and data controls, personnel security awareness levels, fraud and social engineering control levels, computer and telecommunications networks, wireless devices, mobile devices, physical security access controls, security processes, and physical locations such as buildings, perimeters, and military bases. The OSSTMM focuses on the technical details of exactly which items need to be tested, what to do before, during, and after a security test, and how to measure the results. New tests for international best practices, laws, regulations, and ethical concerns are regularly added and updated.

OSSTMM is publicly available for downloading. If followed, OSSTMM ensures that a thorough penetration testing has been undertaken. It also comes with Report Requirements Templates, to assist in the creation of final reports, and a legal penetration testing checklist, containing features to consider, such as privacy and protection of information and authorization for the test. Note that OSSTMM does not give instructions on how to accomplish the penetration testing or what tools to use for it; there are numerous sites on the Internet and books for this task, along with institutions and companies that will happily charge you to attend their seminars and get (a portion of) this knowledge.

Open Source or Proprietary Tools?

Security-related tools exist in both OSS and commercial platforms. Most of the commercial tools are generally more professional looking; however, keep in mind that these are difficult or impossible to modify to fit your needs, and their cost is often significant. Moreover, there are no commercial tools available for several tasks. Also, commercial tools are often created after OSS tools have been available for some time, and therefore such tools lag in the technologies they use. Typical examples of this state of affairs are current WEP analysis and cracking tools. Many OSS security-related tools are maintained by a large team of people, and hundreds of developers contribute to the project. Generally, OSS tool updates are more frequent and signatures for vulnerability assessment tools for the newly discovered vulnerabilities are added soon after they are publicly available. In this area the reflexes of the OSS community appear to be far quicker, and therefore the best tools for penetration testing are not commercial.

What Is a Linux Live CD?

Linux live CDs are Linux systems based on a certain distribution that operate from the distribution CD-ROM without the need to set up the system and without the use of the local hard drive. They perform automated hardware configuration with great success. As a result, within a few minutes from booting, you'll have in front of you a full graphical Linux environment, with all the peripherals identified and a number of preinstalled programs ready to be used. One category of Linux live CDs targets security. Most of those CDs are based on Knoppix or Slax distributions. (Knoppix is a distribution based on Debian, whereas Slax is based on Slackware.)

Alternatives

Live CD distributions for security can be split into the following categories: Penetration Testing, Forensics, and Secure Desktop. The Forensics category focuses on tools for the noninvasive study and retrieval of data

from various types of file systems, whereas the Secure Desktop distributions focus on programs and servers providing secure protocol implementations. Penetration Testing live CDs include programs for enumeration, network scanning and analysis, vulnerability assessment, and exploitation of security vulnerabilities.

A system for penetration testing requires a lot of work to set up, as it involves gathering the programs, installing them, and keeping them up-to-date. A live CD for penetration testing, such as the ones that we will examine here, saves you this effort.

Typically a penetration testing CD will contain the following:

- attack and penetration testing tools
- enumeration tools
- tools for scanning and network port analysis
- vulnerability scanners targeting known problems
- CIFS (SMB) scanners
- sniffers and network analyzers
- tools for the exploitation of common vulnerabilities (e.g., Metasploit Framework, Exploit Tree)
- HTTP proxy tools
- fuzzer tools
- tools for router scanning and exploitation
- tools for spoofing and session hijacking
- tools for password cracking and brute-force attacks

Let's go through some of the available live CDs for penetration testing. You can locate the live CDs in the security category of the frozentech list.[2] All distributions comprise a basic set of penetration testing tools (nmap, nessus, nikto, Metasploit Framework) plus some additional tools to make the system more functional, such as editors, Web browsers, and image viewers. You can see a summary of the features of some prominent distributions in Table 1.

Our personal favorite is the Auditor security collection [3]: It includes all the tools we listed, and more. What we like most about Auditor is the organization of the programs into separate categories, its orientation toward professional administrators, and its cutting-edge functionality. In the wireless sector, the Auditor truly shines, coming with the most complete tool collection for wireless network penetration testing. Some of those programs, such as the wireless LAN scanner Kismet, are notorious for their time-consuming and difficult installation; with Auditor this functionality comes out-of-the-box. Furthermore, Auditor uniquely incorporates tools for Bluetooth penetration testing.

Although some tools are missing from Auditor, with a little additional work an installed system can be transformed into a state-of-the-art base for penetration testing. For example, tools we found missing from Auditor are those for database auditing, for Novell Netware auditing, and for SMB and Kerberos sniffing. Some of these tools exist for Linux, whereas others can operate through Wine. Furthermore, it would be desirable if the system had, by default, read/write capabilities for NTFS file systems. In addition, one could add the Achilles and Spike Web interception proxies; apart from their other capabilities, these automatically test Web applications for buffer overflows and SQL injection.

From the other distributions that we examined we found Whax [4] and KCPentrix [5] most interesting. Both distributions include features that Auditor lacks. For example, Whax contains snort accompanied with acid

and other front-ends, as well as tools for vulnerability enumeration through the so-called Google hacking techniques. In the vulnerability scanners category, Whax has modules for the scanner Retina and Foundstone tools operating through Wine (both Windows tools). Furthermore, Whax includes tools for database auditing: for instance, Absinthe for blind SQL injection, and other tools for auditing Oracle and Cisco systems. Beyond the Metasploit Framework, an advanced open-source platform for developing, testing, and using exploit code, Whax includes Exploit Tree, a properly supported exploit source code base with an update capability. In addition, Whax contains several exploit collections for client-side attacks: vulnerabilities for Internet Explorer as well as exploit archives from the securityfocus.com, packetstormsecurity.com, and milworm.com sites. Both Whax and KCPentrix are founded on Slax and therefore share many features, with Whax offering slightly more material.

The Phlak [6] live CD consists of only a few tools. What impresses us in Phlak is its accompanying security-oriented documentation, which is well organized in different categories. We found this to be very useful and think that other distributions could benefit from adopting this approach. For example, OSSTMM could be included on a security-related live CD.

	GUI	*System Apps*	*Installation Program*	*Vulnerability Scanners*	*Exploit Tools*	*Version in 2005*	*Documents/ Penetration Testing Material*	*Wireless Pen*	*Bluetooth Pen*
Auditor	Y	Y	Y	Y	Y	Y	N	Y	Y
Whax	Y	Y	Y	Y	Y	Y	N	Y	N
KCPentrix	Y	Y	N	Y	Y	Y	N	Y	N
Phlak	Y	Y	Y	Y	Y	Y	Y	Y	N
Knoppix-std	Y	Y	Y	Y	N	N	N	Y	N

TABLE 1. DISTRIBUTION COMPARISON TABLE

Penetration Testing

Often the penetration testing process is presented as a mixture of science and art. Furthermore, complete penetration testing involves something more than the simple execution of various vulnerability scanners targeting some systems: The penetration tester aims at tracing *all* the possible violation pathways, by following a well-defined methodology.

Even if the penetration testing results depend on the knowledge and skills of the penetration tester, there are some tasks that are most customarily followed. Usually, you will initially enumerate the systems or the networks that are to be tested, to obtain basic information about them, for example IP address ranges, gateways, and administrator names. Subsequently, through port scanning, you will locate open ports and services that are running on them. Any network service is a potential door to the system. Services that currently run may be vulnerable to a known vulnerability, something that a vulnerability scanner will show, but they can also be traced manually if you get a connection to an open port, read the banner, and afterward check if the service version is vulnerable to some flaw.

Most services will reveal their version from a banner with little effort, but even those tailored not to reveal such information can be tricked sometimes. It is important to locate all existing shares in Windows systems or NFS exports in UNIX. With brute-force tools, you can try to crack pass-

words that give access to shares or to the system, through SSH, FTP, Web protocols, webmin, or another service. By using a sniffer you can see unencrypted protocols (a formerly common and controversial pastime at USENIX conferences), as well as passwords or other sensitive data that pass through the network. For example, a few years ago, one of us used a sniffer to demonstrate to the public that sensitive data used in a particular setup of a popular e-government application were being transmitted in plaintext form. You can also use Ettercap and Dsniff to perform more sophisticated attacks, utilizing somewhat esoteric techniques, such as ARP spoofing for sniffing through switches. Several other tools that are incorporated in Auditor allow you to test network security and to locate risky setups through spoofing, traffic injection, or DHCP flooding.

When you locate vulnerabilities, you will have to try to exploit them before documenting possible solutions, to ensure that you don't report any false positives or false negatives. For example, an application may be lying about its version, or it may have been configured with a workaround to avoid a particular vulnerability. This is where tools like the Metasploit Framework come in. These tools allow you to avoid false positives and directly check for security gaps. In addition, with such tools you can demonstrate the actual problems, because sometimes system administrators know of certain problems in their network, but they fail to address them, in the mistaken belief that their network is not at risk.

In light of the fact that in many networks Web applications—which are most probably supported by a database—house valuable assets, you'll need to test them separately for how they behave on unexpected input, SQL injection, and other attacks. You could perform this job using tools such as Nikto, Spike, Achilles, or Paros.

Discussion

Obviously, these tools are extremely powerful and in the hands of unauthorized people they cause many problems and chaos on a network. Some may claim that distributions such as Auditor make it easier for script kiddies and other wrongdoers to accomplish their attacks. However, nowadays anyone with a browser can easily find information about the programs Auditor contains; try, for example, Googling for "dhcp flooder." Script kiddies would require some additional effort to install them; eventually though, the tools will work for them.

Conclusions

With a live CD like Auditor you as a system administrator could run Nessus periodically on your systems to check whether there are any security-related problems, or you could use it as a base system for a more complete penetration test. Most of the live CDs we examined allow you to install tools not included in the distribution, and some of the tools support the automated downloading of updates. Both features will help you keep your penetration testing system up-to-date. When the time for downloading the updates becomes excessive, just burn a CD with an updated distribution. Finally, keep in mind that these distributions are maintained by unpaid volunteers; don't forget that these projects depend on contributions from our community for maintenance and improvements.

REFERENCES

[1] Open Source Security Testing Methodology Manual (OSSTMM): http://www.osstmm.org.

[2] Frozentech list with live CDs for security: http://www.frozentech.com/content/livecd.php?pick=All&showonly=Security&sort=&sm=1.

[3] Auditor security collection: http://www.remote-exploit.org/index.php/Auditor_main.

[4] Whax: http://www.iWhax.net.

[5] http://www.kcpentrix.net/Site/.

[6] Phlak: http://www.phlak.

The Fund to Establish the John Lions Chair in Operating Systems at the University of New South Wales

USENIX announces the creation of a matching fund to establish the John Lions Chair in Operating Systems at the University of New South Wales.

The University of New South Wales is establishing an endowed Chair to recognize the enormous contribution made by John Lions to the world of computing. USENIX will match up to $250,000 in donations made through USENIX, now through December 31, 2006. To donate, see below.

The Chair, to be called the John Lions Chair in Operating Systems, will enable an eminent academic to continue the John Lions tradition of insightful and inspirational teaching in operating systems. The creation of the Chair will perpetuate the John Lions name, and new generations of students will benefit from his legacy.

HOW DO I DONATE TO THE JOHN LIONS FUND?

USENIX will match your donation to the John Lions Fund, now through December 31, 2006. Donations can be made by sending a check, drawn on a U.S. bank and made out to the USENIX Association, to:

John Lions Fund
USENIX Association
2560 Ninth St., Suite 215
Berkeley, CA 94710

or by making a donation online at https://db.usenix.org/cgi-bin/lionsfund/donation.cgi.

Your contribution may be tax-deductible as allowed by law under IRS Code Section 501(c)(3). Check with your tax advisor to determine whether your contribution is fully or partially tax-deductible.